REMARKS

Applicants respectfully request that the above-identified application be reexamined.

The Office Action dated December 13, 2005, rejected Claims 1-54. Claims 1-54 were rejected under 35 U.S.C. § 102(e) as being anticipated by Wason et al., U.S. Patent No. 6,701,383. Applicants respectfully disagree for the reasons set forth below. Claims 1, 6, 9, 10, 13, 16, 33, 34, 39, 41, 46, 47, 48, 49, 50, and 53 have been amended to improve grammar and correct antecedent basis errors.

Prior to discussing in detail why applicants believe that all of the claims in this application are allowable, a brief description of applicants' invention and a brief description of the teachings of the cited and applied references are provided. The following discussions of applicants' invention and the cited and applied references are not provided to define the scope or interpretation of any of the claims of this application. Instead, these discussions are provided to help the U.S. Patent and Trademark Office better appreciate important claim distinctions discussed thereafter.

Applicants' Invention

The present invention is generally directed towards a media player hosted by a web browser. More specifically, the invention is directed towards a system and a method for integrating generic media playback components into a web browser and exchanging command and status information between the web browser and the media player. According to one exemplary embodiment, a software architecture allows an Internet browser to host a generic media player. The media player may be any software component that defines a playing state, such as playing, pause, stop, and the like. In one specific exemplary form, the software architecture describes the media player, renders a media player interface, schedules downloads and presentation times, and synchronizes media playback with a timing representation created by the web browser. According to another exemplary aspect of the invention, the media player interface includes various forms, such as a player lifetime management interface; a timing interface that is used to exchange timing information between content, the media player, and the web browser hosting the media player; a playback control interface that is used to control media playback; a rendering interface that is used to render media content; and a playback synchronization interface that is used to communicate timing information between the media player and the web browser. Playback control interfaces and rendering interfaces are generally directed towards object management. Timing interfaces and playback synchronization interfaces are directed towards timing the synchronization of multimedia content. In accordance with other exemplary aspects of the invention, a player-hosting peer software component manages player-hosting in the web browser. The player-hosting peer *negotiates playback state and rendering status* between the web browser and the media player. The player-hosting peer keeps the web browser and *one or more media players* coordinated and synchronized while displaying multiple potentially disparate types of content that may be incorporated into a single document displayed in the web browser.

In accordance with another exemplary aspect of the invention, communication channels are provided for informing media playback components about a web browser's timing. For example, a web content author may use *different embedded playback components* in a *single document*, each having different notions of time.

In summary, this invention is directed towards a software architecture that provides coordinated and synchronized hosting of multiple media players and other content source in a web browser.

Wason et al., U.S. Patent No. 6,701,383 B1 ("Wason")

Wason is generally directed towards an extensible software framework with an overlaying abstraction layer providing a cross-platform interface (abstract). More specifically, Wason discloses a synchronization abstraction layer ("SAL") providing a uniform interface between the framework and one or more plug-ins (Col. 2, lines 26-27). The SAL synchronizes itself and other plug-ins to a timeline provided by the underlying software framework. The plug-ins interact with the underlying framework through the SAL (Col. 2, lines 30-35). Plug-ins are software extension modules that are dynamically loaded into computer memory at run-time (Col. 1, lines 57-60). SAL is an extension module (Col. 3, lines 61-62). Wason et al. discloses a content framework that insulates content and plug-in developers from details and differences in hardware platforms, so that the *same* content *or plug-in* can *run on different desktop platforms*, for example, PC, Macintosh, and Linux (Col. 2, lines 51-55). This is in contrast to the present invention where multiple ***different software components***, such as media players, are synchronized and coordinated by the player-hosting peer software component to run on the ***same*** browser.

In summary, Wason et al. is directed towards a software framework that enables a single plug-in to work on multiple platforms. In contrast, the present invention is directed to a framework that enables different multiple plug-ins to work on the same web browser platform.

Rejection of Claims 1-54 Under 35 U.S.C. § 102(e)

As noted above, Claims 1-54 were rejected under 35 U.S.C. § 102(e) by the Office Action "as being anticipated by Wason." Applicants respectfully disagree. Amended independent Claim 1 recites, in its entirety:

> 1. A system for synchronizing playback of media content with other content or with host computer time information, the system comprising:
>
> a web browser for providing a timing representation to a media player;
>
> a media player *including* a first interface for object management and a second interface for exchanging timing and synchronization information with the web browser; and
>
> a player-hosting peer within the web browser for *negotiating a playback state and a rendering status* between the web browser and the media player.

(Emphasis added.)

Wason et al. does not teach or suggest a "media player including a first interface for object management and a second interface for exchanging timing and synchronization information with the web browser." Wason et al. discloses RealPlayer 301 having three extension modules including SAL 310, RealVideo 302, and RealText 303. Those skilled in the art appreciate that a software component including an interface is different from a software component having extension modules or plug-ins. An interface is a set of functional specifications, such as an application programming interface (API), that provides certain functionalities to other external software components; whereas a plug-in is a self-contained software component that is dynamically included in another software module. Hence, Wason et al. does not teach or suggest a media player implementing an interface, let alone a first interface for object management and a second interface for exchanging timing and synchronization information. Wason et al. does not teach or suggest a player-hosting peer within the web browser for negotiating a playback state and a rendering status *between the web browser and the media player*. Wason et al. discloses a synchronization abstraction layer (SAL), which provides synchronization *between plug-ins and the underlying framework* (emphasis added; Col. 2, lines 31-33). As noted above and as will be appreciated by those skilled in the art, a media player is not a plug-in, but rather a media player is a software component that may include plug-ins at run-time. Additionally, Wason et al. does not teach or suggest *negotiating* a *playback state* and a *rendering status* between the browser and the media player. Wason et al. discloses that "*[w]hen the user clicks* on a particular heading within the table, RealTOC 314 sends the 'time' associated with the heading to the 'seek' function of RealPlayer 301 through JVM 313 and SAL 310." (Col. 5, lines 66-67, to Col. 6, lines 1-2). Those skilled in the art appreciate that in a software protocol negotiation includes autonomous actions taken by a software entity involved in the negotiation. Wason et al. does not teach or suggest *negotiating a playback state*. Wason et al. discloses that, *when the user clicks* on a particular heading, RealTOC 314 invokes the seek function *in response to the user's action*. Therefore, SAL does not *negotiate* a playback state between the browser and the media player. The SAL simply performs the function initiated by the user. Therefore, amended independent Claim 1 is submitted to be allowable for at least the reasons presented above.

Claims 2-38 depend from Claim 1 and are submitted to be allowable for at least the same reasons as presented above with respect to Claim 1. Claims 2-38 are submitted to be allowable for additional reasons. For example, Claim 3 recites, "the media player notifies the player-hosting peer of media player state changes." Wason et al. does not teach or suggest a media player that notifies the player-hosting peer of media player status changes. Wason et al. discloses a RealTOC object 314 that *launches* a seek function. As another example, Claim 36 recites that, "the other content includes advertising or other commercial content synchronized with at least one portion of the media content." Wason et al. does not teach or suggest other commercial content synchronized with a portion of the media content.

Amended independent Claim 39 recites, in its entirety:

> 39. A method of synchronizing playback of media content with other content or with host computer time information, the method comprising the steps of:

providing a timing representation to a media player;

providing a first media player interface for object management and a second media player interface for exchanging timing and synchronization information with a web browser;

issuing commands from the web browser to the media player, the commands being directed to media player operations other than, and in addition to, instantiation of the media player; and

notifying the web browser of media player state changes.

(Emphasis added.)

As discussed above with respect to Claim 1, Wason et al. does not teach or suggest providing a *first media player interface* for object management and a *second media player interface* for exchanging timing and synchronization information. Wason et al. discloses a RealPlayer 301 having three extension modules SAL 310, RealVideo 302, and RealText 303. As discussed above with respect to Claim 1, an extension module is not the same as an interface. Additionally, Wason et al. does not teach or suggest *notifying* the web browser of *media player state changes*. Wason et al. discloses a RealTOC object 314 that launches a seek function *in response to a user clicking* on a particular heading. Therefore, independent Claim 39 is submitted to be allowable for at least the reasons presented above.

Claims 40-54 depend from Claim 39 and are submitted to be allowable for at least the same reasons discussed above with respect to Claim 39. Claims 40-54 are submitted to be allowable for additional reasons. For example, Claim 52 recites that, "the other content includes advertising or other commercial content synchronized with at least a portion of the media content." As discussed above with respect to Claim 36, Wason et al. does not teach or suggest other commercial content synchronized with a portion of the media content.

CONCLUSION

In summary, applicants respectfully submit that all the claims in this application are clearly allowable in view of the disclosures of Wason et al. As a result, applicants respectfully request that all of the claims remaining in this application be allowed and this application be passed to issue. If the Examiner has any questions, the Examiner is invited to contact applicants' attorney at the number set forth below.

Respectfully submitted,

CHRISTENSEN O'CONNOR
JOHNSON KINDNESS^PLLC

Gary S. Kindness
Registration No. 22,178
Direct Dial No. 206.695.1702

I hereby certify that this correspondence is being deposited with the U.S. Postal Service in a sealed envelope as first class mail with postage thereon fully prepaid and addressed to Mail Stop Amendment, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the below date.

Date: May 12, 2006 Patricia Gubble

FXM:mgp